UNITED STATES PATENT APPLICATION

FOR

Efficient Deletion of Archived Data

INVENTORS:

Axel Herbst
Jan Nolte-Boemelburg

Prepared by:

Blakely, Sokoloff, Taylor & Zafman LLP
32400 Wilshire Boulevard
Seventh Floor
Los Angeles, CA 90025-1026
(408) 720-8300

Attorney's Docket No.: 6570/P057/2003/P005/56US

"Express Mail" mailing label number: EV336583791US
Date of Deposit: November 12, 2003
I hereby certify that I am causing this paper or fee to be deposited with the United States Postal Service "Express Mail Post Office to Addressee" service on the date indicated above and that this paper or fee has been addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450
Carla Vignola
(Typed or printed name of person mailing paper or fee)

(Signature of person mailing paper or fee)
11-12-03
(Date signed)

Efficient Deletion of Archived Data

[0001] This application claims the benefit of U.S. Provisional Application Serial No. 60/507,258, filed September 29, 2003.

FIELD

[0002] Embodiments of the invention pertain to the fields of data processing. More particularly, embodiments of the invention relate to data archival.

BACKGROUND

[0003] In today's corporate world, businesses keep large volumes of various data, such as accounting information, customer information, business specific data. With the growth of these volumes of data, management of databases of the business systems becomes more problematic. In order to minimize access times and costs, some business systems move dormant "read-only" data into dedicated archive systems, which are less expensive than the main database systems. For example, the archived data may be stored on cheaper storing medias, such as tapes and disks. The archiving of data provides businesses with long-term access to various data that may be required to be accessed in the future, for example, in a legal proceeding.

[0004] There are existing archiving system solutions on the market. These archiving system solutions perform archiving operations by selecting data objects from a database and writing the data objects into archive files, upon which, the corresponding data objects are deleted from the database, after ensuring that the data was successfully written into the archive files. Data objects written into a single archive file are processed by a single delete job, i.e. a computer process. Thus, a single delete job is required for every newly created archive file in order to delete the

archived data objects from the database. Moreover, the deletion operations in these archiving systems do not halt until all the data objects written into a single archive file are deleted from the database. The number of the created delete jobs and the duration of execution of the created delete jobs may affect the overall system performance. For example, a large number of delete jobs may affect the allocation of processing resources, if delete jobs are occupying resources necessary for more important system processes. However, in the archiving systems currently utilized on the market, the system managers are not provided with any options to configure the archive system to control the number of delete jobs created, and thus are not able to control allocation of processing resources and overall system performance.

[0005] Alternatively, some existing archiving systems store data to be archived in a marked-up form, for example in an Extensible Markup Language (XML) form, by creating an archived XML object for each data object to be archived. In these systems, there is a delete job created for each XML object, i.e. for each data object to be archived. Thus, the number of the created delete jobs in these systems is even greater than in the systems storing data objects in the archive files, and the above-stated problems are aggravated.

[0006] What is needed, therefore, is a solution that overcomes these and other shortcomings of the prior art.

SUMMARY

[0007] A method and system for deleting archived data are disclosed. Embodiments of the invention include requesting identification keys of archived data objects to be deleted from a database. Embodiments of the invention further include deleting at least one data object identified in response to a request for the identification keys while requesting additional identification keys of data objects to be deleted.

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] The invention is illustrated by way of example and not limitation in the figures of the accompanying drawings, in which like references indicate similar elements and in which:

[0009] Figure 1 illustrates components of an archiving system according to one embodiment of the invention;

[0010] Figure 2 is a flow diagram of archiving process according to one embodiment of the invention;

[0011] Figure 3 is a flow diagram of a data deletion process according to one embodiment of the invention; and

[0012] Figure 4 illustrates an exemplary processing system according to one embodiment of the invention.

DETAILED DESCRIPTION

[0013] A method and apparatus for deletion of archived data are described. Note that in this description, references to "one embodiment" or "an embodiment" mean that the feature being referred to is included in at least one embodiment of the invention. Further, separate references to "one embodiment" in this description do not necessarily refer to the same embodiment; however, neither are such embodiments mutually exclusive, unless so stated and except as will be readily apparent to those skilled in the art. Thus, the invention can include any variety of combinations and/or integrations of the embodiments described herein.

[0014] It will be appreciated that the term "data object", as used herein, means one or more records of business data. The term "record", as used herein, means a compilation of business data, for example, a document, a portion of a document, several documents, etc. For example, a record may be a name and a phone number from a phone book, and the data object may be a compilation of records of a sales department in an organization.

Exemplary Architecture

[0015] Figure 1 illustrates components of the archiving system according to one embodiment of the invention. A data archival module 100 includes a writing module 103 to write data objects to be archived into archived data files 155. The data archival module 100 also includes identification (ID) manager 105 for assigning and deleting identification keys (IDs) to data objects to be archived. The identification keys are stored in an ID store database 115. The data archival module 100 includes a lock engine 110 for locking data objects selected for archival from subsequent pick requests, as described below. A delete module 130 includes a delete program 140

deleting the selected data objects from the main database 160. The delete program 140 includes a request engine 135 requesting IDs of data objects to be deleted. The functions of these and other components of the invention are described in detail below.

[0016] It will be appreciated that the components of the invention may be distributed among several processing systems, or may be located on a single processing system. Additionally, a server-client configuration may be used, wherein some components of the invention reside at a server machine and some component of the invention reside at a client machine.

Methodology

[0017] With these concepts in mind, embodiments of the invention may be further described with reference to Figure 2. At 200 the system administrator specifies data objects to be archived. The system administrator may specify the data object to be archived via an interface, or via command-lines. For example, the system administrator may select business documents for the fiscal year 1998-1999 to be archived. The writing module 103 selects data objects corresponding to the business documents for the fiscal year 1998-1999 to be archived. Prior to writing the selected data objects into the archived data files 155, the writing module 103 notifies the ID manager 105 of the data objects to be deleted from the main database 160. Upon receiving the notification, the ID manager 105 assigns IDs to the data objects to be deleted at 210. Once the data objects are successfully written into the archived data files 155 at 220, the ID manager at 230 marks the IDs of the data objects as available for deletion. The IDs of the data objects to be deleted may be stored in an ID Store 115 relational table or database. It will be appreciated that the IDs may be assigned to the data objects after the data objects are written into the archived data files. It will

also be appreciated that a single archived file may include several archived data objects or a single data object in a marked up form, for example an XML form.

[0018] In one embodiment, a plurality of delete jobs are executing in parallel, the number of which is specified by the system administrator. Prior to creating delete jobs, the system administrator may determine whether the system resources have to be allocated to more important processes, in which case, the system administrator specifies the number of delete jobs to be performed accordingly.

[0019] Functions of a delete program are now described according to one embodiment of the invention. Once the data objects are marked as available for deletion, the delete module 130 may delete the data objects from the main database. In one embodiment at 300 of Figure 3, the request engine 135 requests IDs of data objects to be deleted from the data archival module 100 by issuing a command, for example, a PICK command. In one embodiment, the request engine continuously issues requests to the data archival module 100 for IDs of data objects to be deleted until a response returns no further IDs. For example, upon deletion of data objects identified in response to a first PICK command, a second PICK command is issued, and if more IDs are returned in response to the second PICK command, the deletion operation is performed, otherwise the delete program is terminated. It will be appreciated that delete jobs may not delete an equal number of data objects, for example, a faster executing delete job may go through a greater number of PICK command iterations than a slower executing delete job, causing the faster executing delete job to delete more data objects than the slower executing delete job.

[0020] At 310 the data archival module selects unlocked IDs from the ID Store 115 and provides the selected IDs to the delete module 130 for deletion in response to the PICK request. In one embodiment the IDs are selected from the ID Store 115

based on a system administrator configurable parameters. For example, the system administrator may specify a maximum number of IDs to be returned in response to a single PICK command. In another embodiment, the IDs are selected based on a relationship between the subject matter of the data objects. For example, the data objects may be grouped according to logical partitions known to the archiving system. For instance, if all the financial documents for the fiscal year 1998-1999 are to be archived, the documents referring to the same account may make up a partition. In this embodiment the data archival module 100 provides the delete program 140 with IDs that belong to the same partition in response to the PICK request.

[0021] At 320 upon the selection of IDs of data objects to be deleted, the lock engine 110 locks the IDs and/or the data objects to ensure that no PICK commands are issued again against these Ids and/or data objects. Locked IDs indicate that the data objects corresponding to the IDs are being currently deleted by other delete jobs and these locked IDs are not selected for deletion in response to subsequent PICK requests.

[0022] According to one embodiment, at 330 the delete program 140 sends a request to the data archival module 100 for the content of data objects corresponding to the received IDs. Alternatively, the delete program 140 may obtain the contents of the data objects to be archived by directly accessing the archived data files 155. Yet, alternatively, the delete program 140 may deduce the location of the content of the data objects to be archived from the IDs assigned to the data objects. For example, IDs may represent location path of data objects, such that if location of the data objects to be deleted is /root/my_system/session4711/, and folder 4711 includes several data objects to be deleted, booking1, booking2, etc., the IDs assigned to these data objects are /root/my_system/session4711/booking1,

/root/my_system/session4711/booking2, etc. Utilizing these IDs the delete module 130 deduces the location of contents of the data objects from the IDs.

[0023] Once the delete program 140 obtains the content of the data objects to be deleted, the delete program 140 deletes the data objects form the main database 160 at 340 upon confirming that the content to be deleted corresponds to the content of the data objects written in the archived data files 155. Upon deletion of the data objects, the delete program 140 at 350 sends a confirmation message to the data archival module 100, to confirm the deletion of the selected data objects and resumes issuing the PICK command as described above. Upon receipt of the confirmation message, the data archival module 100 deletes the identification keys associated with the deleted data objects from the ID store 115. In one embodiment, the data archival module 100 issues a deletion confirmation message to the system administrator notifying the system administrator that the selected data objects were successfully archived and deleted from the main database.

[0024] It will be appreciated that physical processing systems, which embody components of the archiving system described above, may include processing systems such as conventional personal computers (PCs), embedded computing systems and/or server-class computer systems according to one embodiment of the invention. Figure 4 illustrates an example of such a processing system at a high level. The processing system of Figure 4 may include one or more processors 400, read-only memory (ROM) 410, random access memory (RAM) 420, and a mass storage device 430 coupled to each other on a bus system 440. The bus system 440 may include one or more buses connected to each other through various bridges, controllers and/or adapters, which are well known in the art. For example, the bus system 440 may include a 'system bus', which may be connected through an adapter to one or more

expansion buses, such as a peripheral component interconnect (PCI) bus or an extended industry standard architecture (EISA) bus. Also coupled to the bus system 440 may be the mass storage device 430, one or more input/output (I/O) devices 450 and one or more data communication devices 460 to communicate with remote processing systems via one or more communication links 465 and 470, respectively. The I/O devices 450 may include, for example, any one or more of: a display device, a keyboard, a pointing device (e.g., mouse, touch pad, trackball), and an audio speaker.

[0025] The processor(s) 400 may include one or more conventional general-purpose or special-purpose programmable microprocessors, digital signal processors (DSPs), application specific integrated circuits (ASICs), or programmable logic devices (PLD), or a combination of such devices. The mass storage device 430 may include any one or more devices suitable for storing large volumes of data in a non-volatile manner, such as magnetic disk or tape, magneto-optical storage device, or any of various types of Digital Video Disk (DVD) or Compact Disk (CD) based storage or a combination of such devices.

[0026] The data communication device(s) 460 each may be any device suitable to enable the processing system to communicate data with a remote processing system over a data communication link, such as a wireless transceiver or a conventional telephone modem, a wireless modem, an Integrated Services Digital Network (ISDN) adapter, a Digital Subscriber Line (DSL) modem, a cable modem, a satellite transceiver, an Ethernet adapter, Internal data bus, or the like.

[0027] It will be recognized that many of the features and techniques described above may be implemented in software. For example, the described operations may be carried out in a processing system in response to its processor(s) executing

sequences of instructions contained in memory of the device. The instructions may be executed from a memory such as RAM and may be loaded from a persistent store, such as a mass storage device, and/or from one or more other remote processing systems. Likewise, hardwired circuitry or firmware may be used in place of software, or in combination with software, to implement the features described herein. Thus, the invention is not limited to any specific combination of hardware circuitry and software, nor is it limited to any particular source of software executed by the processing systems.

[0028] Thus, a method and apparatus for deletion of archived data have been described. Although the invention has been described with reference to specific exemplary embodiments, it will be evident that various modifications and changes may be made to these embodiments without departing from the broader spirit and scope of the invention as set forth in the claims. Accordingly, the specification and drawings are to be regarded in an illustrative sense rather than a restrictive sense.